

A Modelica Power System Component Library for Model Validation and Parameter Identification

Luigi Vanfretti^{1,2} Tetiana Bogodorova¹ Maxime Baudette¹

1: Smart Transmission Systems Lab. (SmarTS Lab), Electric Power Systems Department
KTH Royal Institute of Technology, Stockholm, Sweden.

2: R&D Division, Statnett SF, Oslo, Norway

Abstract

This paper summarizes the work performed in one of the work-package of the FP7 *iTesla* project. This work consisted in the development of a power system component library for phasor time domain simulation in Modelica.

The models were used to build power system network models, used in experiments for parameter identification. The experiments were carried out with the RAPID toolbox, which has been developed at SmarTS Lab within the same project. The toolbox was written in MATLAB, making use of FMI Technologies for interacting with Modelica models.

Keywords: Power Systems, Phasor Simulation, Modelica, FMI, Parameter Identification, Model Validation.

1 Introduction

1.1 Conventional Power System Modeling and Simulation

Modeling power system components and networks is important for different studies in the planning and operation of electricity networks [1], and the level of complexity in the models depends on the type of studies carried out [2]. Traditional tools for power system modeling are usually tied to a specific time scale, often limiting the applicability and/or validity of the models to a specific kind of studies [3].

The models can be of Electro-Magnetic Transient (EMT) type [4, 5], which is the most detailed type of power system models; of phasor time-domain (phasor) type, which use a simplified representation commonly used in positive sequence phasor time-domain simulation for stability assessment [2]; or could even use a Quasi Steady State (QSS) representation for simplified long term dynamic simulation [6]. To assemble power

system models using different components of specific type and perform simulations, domain-specific tools are traditionally used; for example [7, 8] are used in the case of phasor-time domain simulation.

Other modeling and simulation approaches which are less utilized in the domain is the dynamic phasors approach [9], and mixed EMT & phasor simulation [10, 11].

The reminder of this article focuses on phasor time-domain power system modeling using the Modelica language and the application of Flexible Mockup Interface (FMI) Technologies for model validation and parameter identification.

1.2 Motivation — Modelica, an unambiguous modeling language for Power Systems

Several tools for phasor time domain simulation already exist, but only few of them let the user access the models' code (mainly those that are open source software [12]), lack flexibility in regards to modeling and simulation features provided [2], and have limitations for unambiguous model exchange [13]. This motivated the work presented in this paper which is part of the FP7 *iTesla* project [14], where a power system component library of phasor type models is being developed.

Modelica was chosen as the modeling language for this library as it allows to develop models using a formal mathematical description and because it separates the model from the solver [3]. The models developed in Modelica use an explicit mathematical representation waving out any ambiguity about the model, while enabling further seamless simulation with diverse tools [3].

Modelica is an object-oriented programming language where the parameters of a model are the object's attribute. This facilitates the development of power

system models composed by instances of components previously developed.

To the knowledge of the authors, there have been two previous efforts in the use of Modelica for power system modeling and simulation: the ObjectStab [15] library, and the SPOT [16] library which evolved into the commercial “Electric Power Library” by Modelon AB [17]. While these libraries could have been used for the purposes of this article, there are social aspects of resistance to change [18] that prompted the authors to develop a new library. For example, users of a specific power systems tool are skeptical about other tools different from the one that they use — this is a complex behavior [19] that involves both the change in the human interactions that accompany the use of a specific tool, as well as a concern about the technical aspects of a new technology.

To make a transition where the use of Modelica is accepted by power system practitioners, the authors and other colleagues have used different strategies to overcome the resistance of domain experts. This approach is similar to that in [20]: a first strategy is to promote the use of Modelica in power systems as a medium for unambiguous model exchange [3] which is an Alpha [20] strategy that supports the common goal of model exchange; a second strategy (type Omega [20]) was used to decrease avoidance forces: made a software-to-software validation of each power system component of *trusted* domain-specific power system tools where simulation results between the Modelica library in different tools are appraised against a domain-specific tool [21].

Further details on power system modeling using Modelica as approached in this article are available in [21, 3].

1.3 Exploiting Modelica and FMI Technologies

Modelica models can also be exploited through Functional Mockup Interface (FMI) [22]; a standard for model exchange between different tools supporting and implementing the standard. As such, Dymola [23], OpenModelica [24] and JModelica.org [25] implement the standard for both import and export; as well as co-simulation.

FMI Technologies offer wide possibilities for model re-use within different software tools. Thus, the models developed and used in Modelica are not tied to the development environment, contrarily to traditional tools for power system modeling. This flexibility

was taken advantage of to develop a Rapid Parameter Identification toolbox (RAPID).

1.4 Paper Organization

The remainder of this paper is organized as follows. The power system component library is described in Section 2. Details of two of the models developed by SmarTS Lab and their validation are shown in Section 3 and a model validation experiment using the RAPID toolbox is presented in Section 4. Conclusions are drawn in Section 5

2 Library Structure

The library was developed for power systems phasor time-domain simulation in Modelica. It is comprised by different packages that build its main structure, which is described next.

2.1 Structure

A total of 120 models are categorized into the following upper level packages (see Fig. 1):

- Connectors (2):
The package is comprised by two connectors, one, PwPin, is adopted for treating voltage and current as complex variables. The other, ImPin is a simple connector for real variables.
- Electrical (56):
The main package of the library with all the power system component models for phasor time domain representation. It is divided into several sub-packages.
- Examples (17):
A set of examples using the power system component models.
- Interfaces (2):
Blocks for adding some inputs and outputs to the models.
- NonElectrical (45):
A set of functions extended from the Modelica standard library and adapted with the connectors of this power system components library.

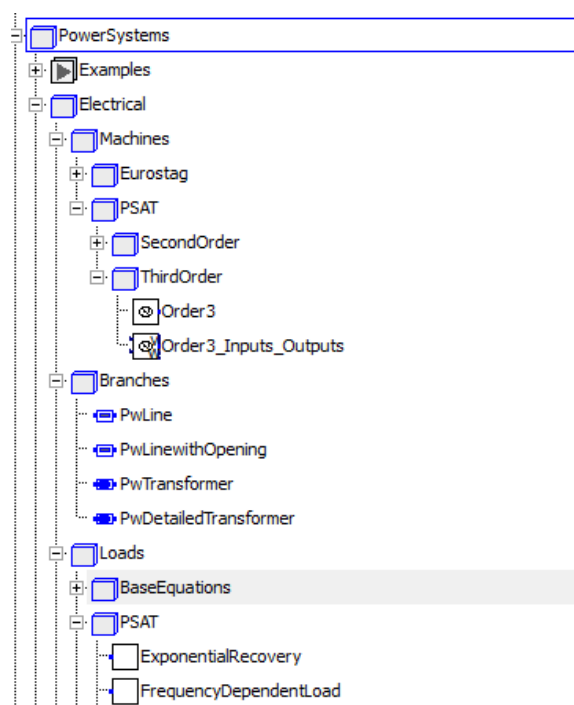


Figure 1. Partial screenshot of the library

2.2 Electrical Components

As a joint effort from the different project participants, the power system components models have mostly been transcribed from the *Eurostag* [8] and PSAT [26] software.

Different components such as transmission lines, transformers, buses and some events (line opening and faults) were developed as generic models. All load models, were implemented following the representation used by PSAT and *Eurostag*. Generator models have also been implemented, including detailed models as used in *Eurostag* as well as different models from PSAT. Additionally some non traditional generators have been modeled including photo-voltaic models (3) and two wind turbines models. These models enable the simulation of complete power system networks with any kind of perturbation.

The models developed at SmarTS Lab are the following:

- Load models from PSAT (Exponential recovery, Frequency dependent, Constant PQ, Voltage dependent, ZIP, ZIP Jimma and Mixed loads)
- An Automatic Voltage Regulator (AVR) from PSAT (Type III)
- A Turbine Governor (Type II) from PSAT
- Synchronous generators (second and third order) from PSAT

- Solar photo-voltaic panel (two simplified models from PSAT and detailed model from KTH)
- Doubly Fed Induction Generator (DFIG) wind turbine (PSAT model and GE model)

These models have been developed together with examples, which served for parameter identification experiments such as the one presented in Section 4.

3 Implementation and Validation of Power System Component Models

All models developed at SmarTS Lab have been created from a reference model taken in another domain specific simulation environment. They have been subject of a Software-to-Software validation against the reference model by comparing simulation results.

In this Section, two of the models developed at SmarTS Lab are presented in details together with their validation results.

3.1 GE Wind Turbine Model

This first study case illustrates the development of a wind turbine generator model taking as reference a MATLAB/SIMULINK implementation of the wind turbine-generator model for grid studies used by General Electric Energy (GE). The wind turbine-generator is Type III, which commonly refers to an induction generator with the rotor winding connected to the grid through a back-to-back converter. The descriptive reference provided by GE Energy can be found in [27].

3.1.1 Modeling

The reference was implemented with function blocks in MATLAB/SIMULINK, organized in three different subsystems (Turbine, Electrical Control and Generator). The system was developed with the voltages (real and imaginary parts) as inputs and currents (real and imaginary parts) as outputs. The system also has an input for the wind speed.

For the Modelica model, the model structure and the block diagram were preserved. The main motivation for this approach was to maintain the existing initialization algorithm used in the reference model. This algorithm is responsible for initialization of, among others, all the integrators in the model.

The resulting model is shown in Fig. 2, where the top-level layer of the model can be observed. The major difference is the integration of the PwPin connector to interface the model to other components, such as

transmission lines. This connector contains two standard variables used for the real and imaginary part of the voltage, as well as two flow variables used for the real and imaginary part of the current.

The Modelica model also implements the initialization algorithm aforementioned. This implementation required to adapt the code to the Modelica syntax by replacing all `for`-loops by `while`-loops, as in Modelica `for`-loops cannot be exited before completion of all iterations.

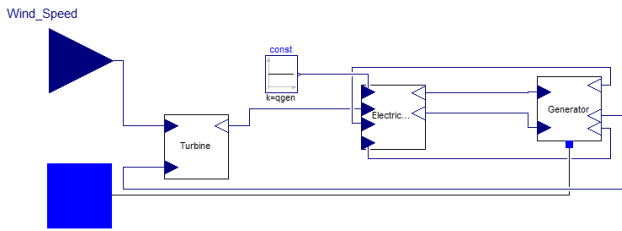


Figure 2. Modelica implementation of the GE Wind Turbine-Generator Model

The original wind turbine reference model was integrated within a test system comprised by a network model, an infinite bus and a wind generator. The network model was built using an admittance matrix, which in the Modelica implementation was replaced by several transmission lines in series. The remainder of the model in Modelica was implemented with components available in the library. A wind generator was additionally developed in Modelica, to create different wind profiles. The resulting test model is presented in Fig. 3.

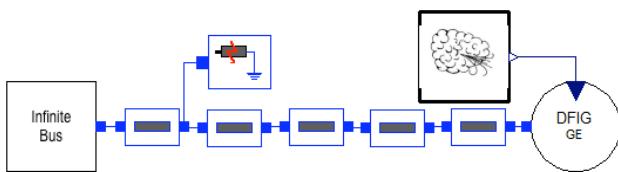


Figure 3. GE Wind Turbine Modelica Test Model

3.1.2 Validation results

Figure 3 presents the test model, which includes the simulation scenario comprised by a gust of wind at $t = 5$ s and a fault at one transmission line at $t = 10$ s.

Both models have the same simulation scenario and are simulated in their respective environment. However, the solver used is different in each environment, a fixed time-step solver ode3 was selected for the reference model, whereas an adaptive time-step solver DASSL was selected for the Modelica model.

A comparison between the simulation of the two models is presented in Fig. 4, where it can be observed that the simulations produce the same response. This validates the implementation carried out in Modelica.

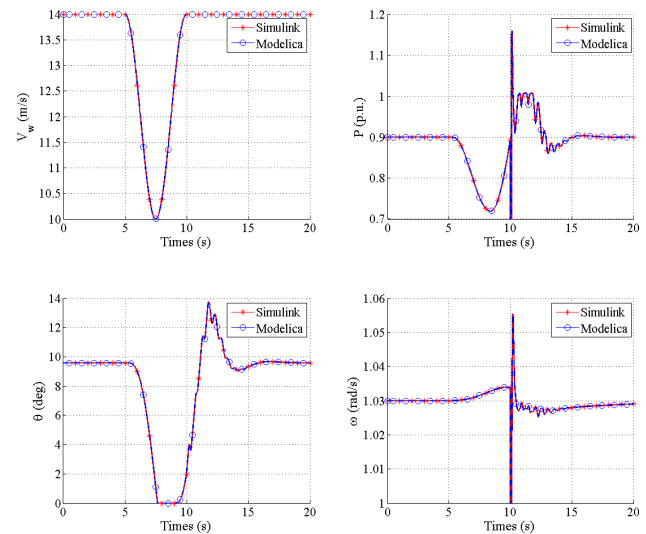


Figure 4. Comparison Modelica vs Simulink for the GE Wind Turbine-Generator Model

However, some differences can be noticed when inspecting the responses more closely. Figure 5 presents an example of this observation. The differences are of a very small order and can be explained by the accuracy of the solvers used in each simulation environment.

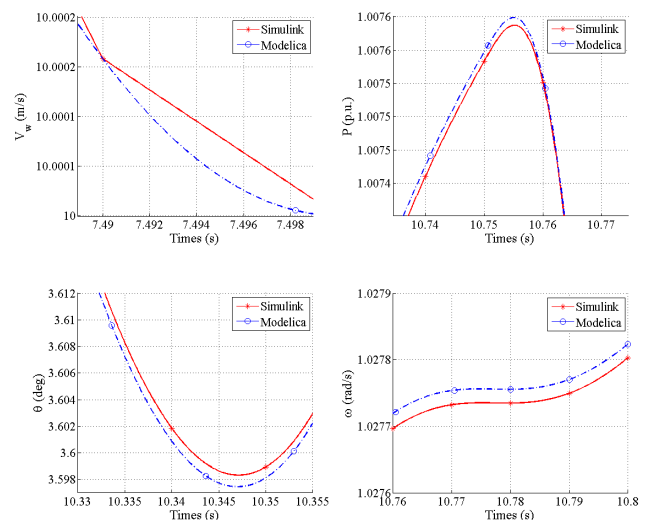


Figure 5. Zoom on comparison Modelica vs Simulink for the GE Wind Turbine-Generator Model

3.2 Hydro Turbine Governors

The second study case illustrates the development of a hydro turbine and governor model. The reference

is a PSAT implementation of type 5 hydro turbine and governor (HTG Type 5) modeled at KTH and included into PSAT. The HTG Type 5 models includes a PI controller combined with servomotor (turbine governor model) and a nonlinear turbine model, its Modelica implementation is shown in Fig. 6.

The hydro turbine and its governor are widely used for control of mechanical power delivered by the generator, and often show complex dynamic behavior. Due to this fact, it is a perfect component to show the advantages of using the Modelica language instead of conventional power system modeling tools for simulation purposes. In this section two Software-to-Software validation cases are presented. The first shows the benefits of using Modelica with respect to PSAT. The second presents an alternative validation approach where the validation of the HTG Type 5 model is carried out in a hybrid SIMULINK-Modelica model exploiting the FMI standard which allows to include an FMU as a part of the original MATLAB/SIMULINK system.

3.2.1 Modeling

The reference model was implemented in MATLAB/SIMULINK using the built-in Control library for the HTG Type 5 and the SimPowerSystems (SPS) library for the single-machine-infinite bus (SMIB) system. A similar system (with a finite load) was modeled in PSAT and taken as a reference for Software-to-Software validation. Next, the Modelica model of the HTG Type 5 and the same system as in PSAT were built in Dymola (Fig. 7).

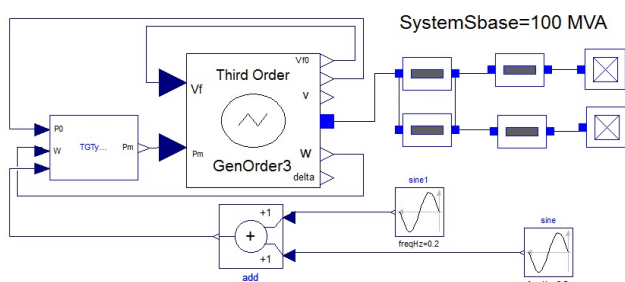


Figure 7. Modelica model of Single machine governed by HTG Type 5 with perturbation

In this the first case, the same initial values are used in Modelica and PSAT, this allows us to carry out the Software-to-Software validation not only in terms of the modeling approach, but also using the same initial conditions.

The types of connectors used in the Modelica model were described above. All the blocks which were

used for the HTG model can be found in the standard Modelica library. For building the hybrid SIMULINK-Modelica model, an FMU file was generated using Dymola to allow simulation in MATLAB/SIMULINK-SPS (Fig. 8). The model was encapsulated into one block, with ability to set up input parameters for the block (Fig. 9) using the FMI Toolbox [28].

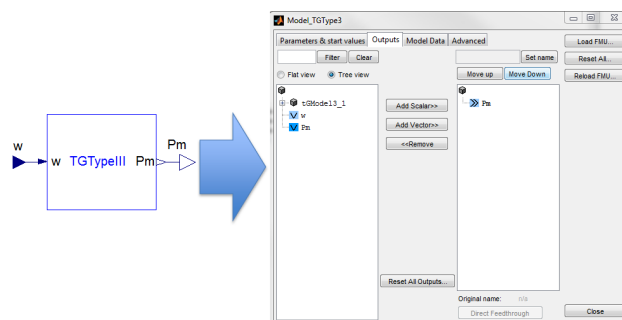


Figure 8. FMU from the Modelica model imported into the FMI Toolbox

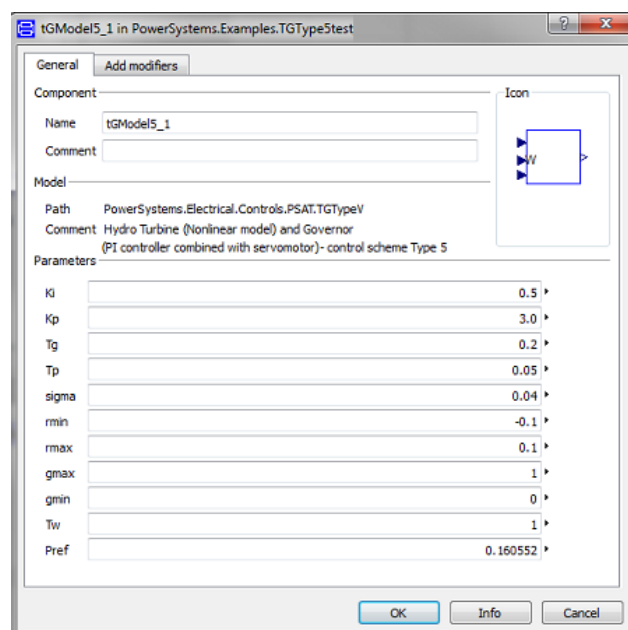


Figure 9. Parameters of HTG Type 5 block

Thus, the HTG Modelica model was used to replace the turbine and governor SIMULINK block in the power system model. This allowed simultaneous simulation and comparison of the response between the SPS models and the Modelica models within the same simulation tool (Fig. 10).

3.2.2 Validation results

For validation of the hydro turbine and its governor the input (speed) and output (mechanical power) were compared (Fig.11). The results show that the “shape”

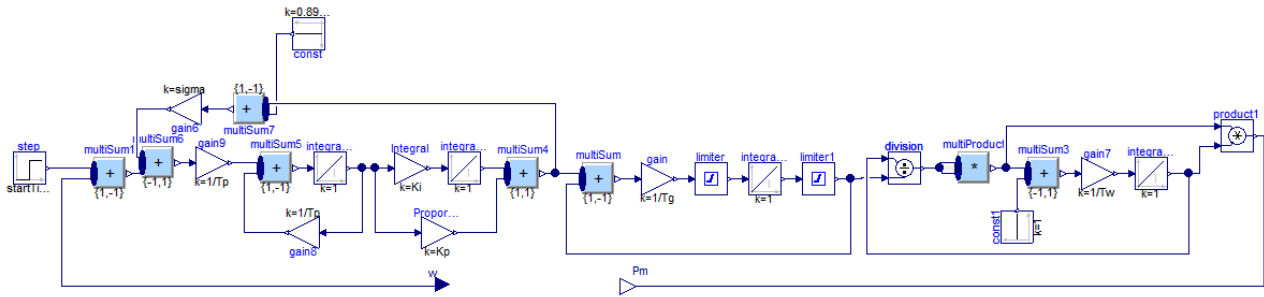


Figure 6. Modelica model of HTG Type 5

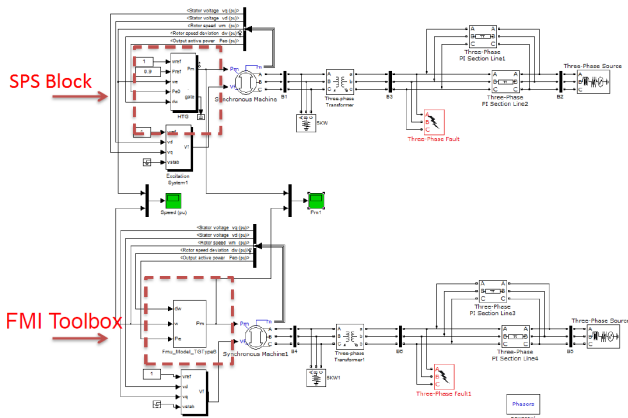


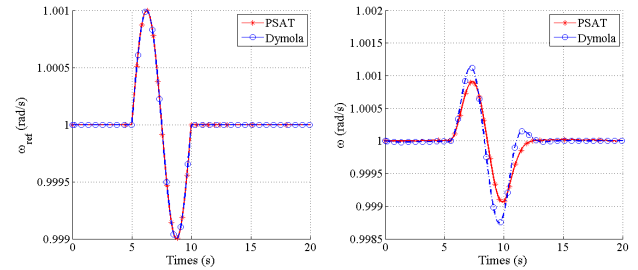
Figure 10. Validation of the HTG model encapsulated using the FMI Toolbox for MATLAB

of the signals match well, but there are some differences. The solver used for simulation with Dymola was DASSL. The differences are due to limitations on the numerical accuracy of the solver in PSAT (a trapezoidal integration with fixed time-step). Please observe that PSAT has a limited number of solvers (trapezoidal integration and forward Euler's method) and that the time-domain simulation methodology follows domain-specific practices not common to general purpose tools.

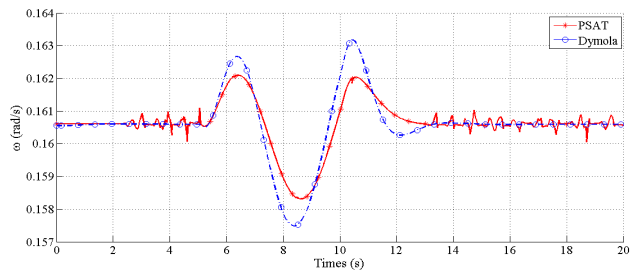
In second validation method, which uses SIMULINK models with the Modelica model embedded in an FMU block and the MATLAB/SIMULINK-SPS reference model, the speed and the output mechanical power were compared (Fig. 12). The results show a very good match between the corresponding outputs.

4 Model Validation using Modelica and FMI Technologies

This power system component library has been developed within the FP7 *iTesla* project with the goal of using the models for power system model validation. The choice of Modelica as modeling language



(a) The input (speed) of the HTG Type 5



(b) The output (mechanical power) of the HTG Type 5

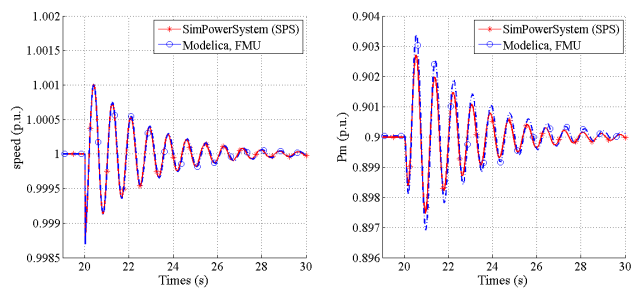
Figure 11. Comparison Modelica vs PSAT for the HTG Type 5's responses with a perturbation at the speed reference w_{ref} .


Figure 12. Validation of the HTG model encapsulated using the FMI Toolbox for MATLAB

for this work was also motivated for its support of the FMI standard, which enabled the development of the RAPID toolbox.

4.1 FMI Technologies

The Functional Mock-up Interface (FMI) standard was created as a tool independent format for model ex-

change and co-simulation. The full specifications of the standard, with a list of the supporting tools, are available in [22]. By opening the possibility of using a model in disparate simulation environments, the standard allows to benefit from functionalities offered by different programs.

As another part of the FP7 *iTesla* project, a parameter identification MATLAB toolbox was also developed. This toolbox was built with FMI technologies and uses the Modelica models presented above. The power system models used are test models including an device to be calibrated against some reference signals. An example of using the toolbox for calibrating the parameters of a generator model is presented next.

4.2 Example

This parameter identification example computes seven of the generator model parameters using reference measurement from the simulation of an EMT model (4 outputs: the voltage magnitude, the rotor speed, and the active and reactive powers). The reference model in this case was built in MATLAB/SIMULINK-SPS. The authors attempted to substitute the complex dynamics of the EMT generator model with a simplified third order generator model. The other components of the system were assumed to be known and modeled with the same components. For this experiment the FMI Toolbox from Modelon AB [28] has been exploited. The reasons for that are the following. First, MATLAB was chosen for the development of the RAPID toolbox. Second, the authors decided to use the Modelica language for power system modeling purposes [3]. The Modelica model with two different test scenarios was constructed (Fig. 13). In both test scenarios the reference signals are perturbed, the nominal torque value in the first and the field voltage in the second.

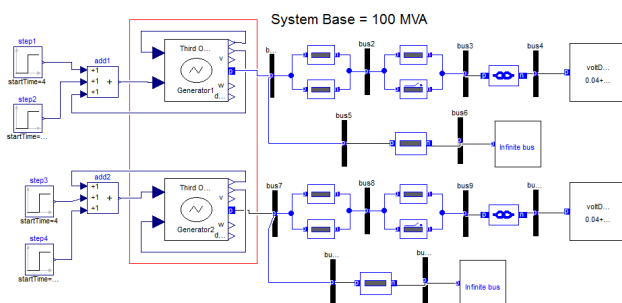


Figure 13. Modelica model used in the parameter identification process (component to be identified bounded in red)

The algorithm of the identification process is the fol-

lowing:

1. Collect measurement data from the reference SIMULINK model.
2. Create the power system model to be identified in Modelica. The Modelica model requires a power flow solution, in this case the results of the power flow were obtained from PSAT.
3. Compile an FMU from the Modelica model in Dymola.
4. Create a SIMULINK model using the FMU block from the FMI Toolbox for MATLAB, see Fig. 14. The only blocks required in order to carry out the simulation are: *FMUme* and *To Workspace*. All data inputs and the scopes are included only to monitor the process interactively.
5. Start the RAPID Toolbox.
6. Provide RAPID with appropriate settings (algorithm, parameters, variables, etc.).

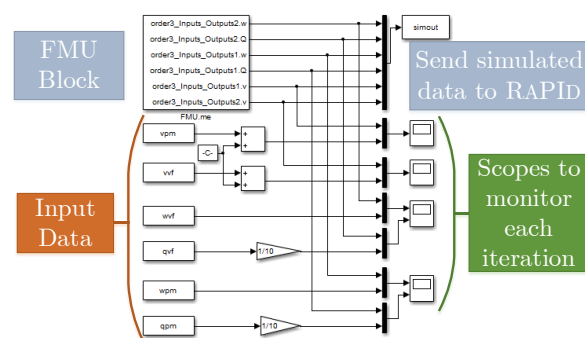


Figure 14. SIMULINK model used by RAPID with the FMU of the Modelica model

4.3 Results

The results of identification experience are presented in Fig. 15 and Table 1. The values of the identified parameters are shown in Table 1.

The numerical values for all the parameters were constrained before running the optimization algorithm to a valid range of real valued numbers, typical of synchronous machines. Thus, the resulting parameters are within practical and realistic values. Figure 15 shows the graphical comparison between the simulations in MATLAB/SIMULINK-SPS and Modelica. It

Table 1. Generator parameter estimation results

Parameter	Value
Armature resistance (R_a)	0.0010156
Direct axis reactance (X_d)	4.2924
Direct axis transient reactance (X'_d)	1.37
Direct axis transient time const. (T'_d)	2.6156
Quadrature axis reactance (X_q)	5.3994
Inertia coefficient (M)	14.9005
Damping ratio (D)	0.0088415

shows that the responses match with an acceptable error. The key here is to remember that a reference signal (from a high order model (MATLAB/SIMULINK-SPS)) is matched to a low-order model. The differences in reactive power (Fig. 15(a)) and in the voltage magnitude (Fig. 15(b)) are acceptable if one takes into account the difference in complexity of the mathematical presentation of high order dynamics by the simplified model. In order to validate the identification results, the simulations in both MATLAB/SIMULINK-SPS and Modelica were repeated using perturbations two-times larger than the original experiments used for identification. Similar results as those reported above were obtained.

5 Conclusion

This article introduced the work performed by SmarTS Lab and other collaborators within the FP7 *iTesla* project for developing a library in Modelica for power system component models. This library is already quite versatile as it enabled to carry out model calibration experiments within work package three of the FP7 *iTesla* project.

But the library is still not complete, additional models are being developed at KTH SmarTS Lab to continue populating it. To this aim, the method for developing and validating new models described in Section 3 is used. As such, the newly developed models can be integrated in the library after a successful Software-to-Software validation step.

The overall motivation for the development of such library in the Modelica language was briefly presented in Sections 1 and 4. The potential of model exchange across different tools is indeed of great importance. This is especially true for future power system tools [13], which will require more advanced simulation methods that the currently available in domain-specific tools. For example, unambiguous model exchange will allow for the development of model-driven

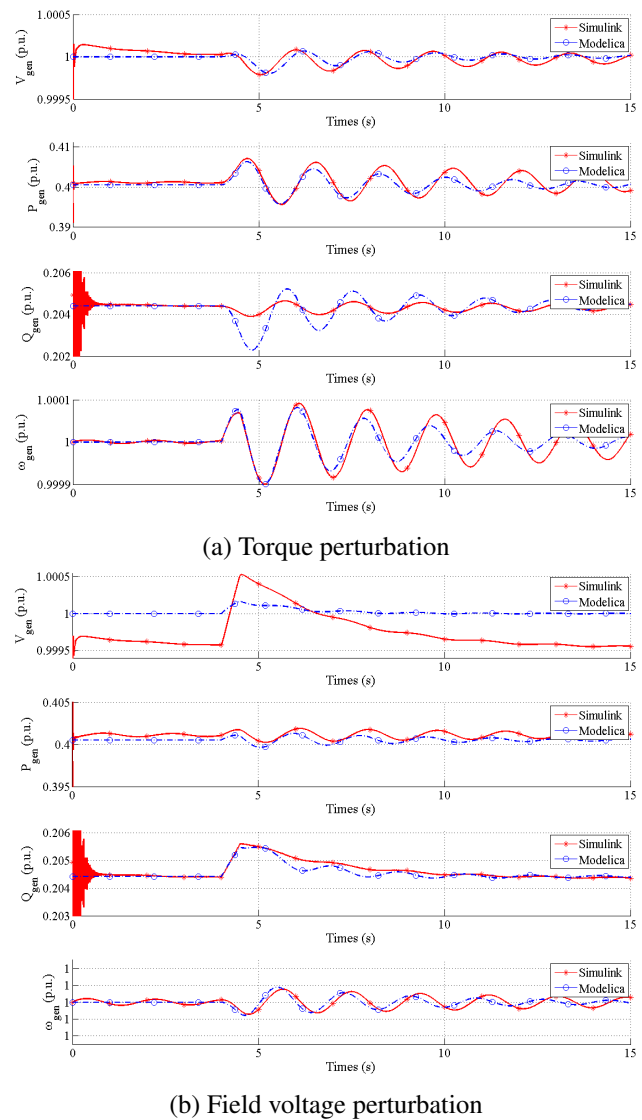


Figure 15. Comparison between the reference (Simulink) and the identified (Modelica) model responses with a perturbation at $t = 4\text{sec}$.

design of wide-area controls, which will require the use of a cyber-physical modeling language and appropriate solvers.

Acknowledgement

The support Dr. Juan Sanchez-Gasca at GE Energy in providing a MATLAB/SIMULINK model of a wind turbine-generator is gratefully acknowledged.

L. Vanfretti was supported by Statnett SF, the Norwegian Transmission System Operator, the STANDUP for Energy collaboration initiative, the EU funded FP7 *iTesla* project and Nordic Energy Research through the STRONG²rid project.

T. Bogodorova was supported by the EU funded FP7 *iTesla* project.

M. Baudette was supported by the EU funded FP7 *iTesla* project and by Statnett SF, the Norwegian TSO.

References

- [1] M. Butts and H. S. Smith, "Dynamic oscillations predicted by computer studies," *IEEE Computer Applications in Power*, vol. 4, no. 1, pp. 47–51, 1991.
- [2] F. Milano, *Power System Modelling and Scripting*. Springer, 2010.
- [3] L. Vanfretti, W. Li, T. Bogodorova, and P. Panciatici, "Unambiguous Power System Dynamic Modeling and Simulation using Modelica Tools," *IEEE PES General Meeting*, 2013.
- [4] J. Mahseredjian, V. Dinavahi, and J. Martinez, "Simulation Tools for Electromagnetic Transients in Power Systems: Overview and Challenges," *IEEE Transactions on Power Delivery*, vol. 24, no. 3, pp. 1657–1669, 2009.
- [5] H. Dommel, "Digital computer solution of electromagnetic transients in single-and multiphase networks," *Power Apparatus and Systems, IEEE Transactions on*, vol. PAS-88, no. 4, pp. 388–399, 1969.
- [6] T. Van Cutsem, Y. Jacquemart, J.-N. Marquet, and P. Pruvot, "A comprehensive analysis of mid-term voltage stability," *IEEE Transactions on Power Systems*, vol. 10, no. 3, pp. 1173–1182, 1995.
- [7] *PSS/E Program Operation Manual*, Power Technologies, Inc., Dec. 1996.
- [8] Eurostag. Accessed: 2013-12-04. [Online]. Available: <http://www.eurostag.be>
- [9] S. Sanders, J. Noworolski, X. Liu, and G. C. Verghese, "Generalized averaging method for power conversion circuits," *Power Electronics, IEEE Transactions on*, vol. 6, no. 2, pp. 251–259, 1991.
- [10] V. Jalili-Marandi, F. Ayres, C. Dufour, and J. Belanger, "Real-time electromagnetic and transient stability simulations for active distribution networks," in *International Conference on Power System Transients (IPST)*, 2013.
- [11] F. Plumier, C. Geuzaine, and T. V. Cutsem, "A multi-rate approach to combine electromagnetic transients and fundamental-frequency simulation," in *International Conference on Power System Transients (IPST)*, 2013.
- [12] F. Milano and L. Vanfretti, "State of the art and future of oss for power systems," in *Power Energy Society General Meeting, 2009. PES '09. IEEE*, 2009, pp. 1–7.
- [13] E. Lambert, X. Yang, and X. Legrand, "Is CIM suitable for deriving a portable data format for simulation tools?" *IEEE PES General Meeting*, pp. 1–9, July 2011.
- [14] *iTesla: Innovative Tools for Electrical System Security within Large Areas*. [Online]. Available: <http://www.itesla-project.eu/>
- [15] I. Navarro, M. Larsson, and G. Olsson, "Object-oriented modeling and simulation of power systems using Modelica," in *IEEE Power Engineering Society Winter Meeting*, vol. 1, 2000, pp. 790–795.
- [16] SPOT - Free library providing components to model power systems both in transient and steady-state mode. [Online]. Available: <https://github.com/modelica-3rdparty/SPOT>
- [17] Modelon AB. Electric power library description. [Online]. Available: <http://www.modelon.com/products/modelica-libraries/electrical-power-library/>
- [18] J. Ford, L. Ford, and A. D'Amelio, "Resistance to change: The rest of the story," *Academy of Management Review*, vol. 33, no. 2, pp. 362–377, 2008. [Online]. Available: <http://amr.aom.org/content/33/2/362.abstract>
- [19] P. Lawrence, "How to Deal with Resistance to Change (HBR Classic)," *Harvard Business Review*, Jan. 1969.
- [20] J. Kavanagh, "Resistance as Motivation for Innovation: Open Source Software," *Communications of the Association for Information Systems*, vol. 13, no. 36, 2013.
- [21] T. Bogodorova, M. Sabate, G. León, L. Vanfretti, M. Halat, J. Heyberger, and P. Panciatici, "A Modelica Power System Library for Phasor Time-domain Simulation," in *IEEE PES Innovative Smart Grid Technologies (ISGT)*, october 2013.
- [22] Functional Mock-up Interface. Accessed: 2013-12-04. [Online]. Available: <https://fmi-standard.org/>
- [23] Dymola, Dassault Systems. [Online]. Available: <http://www.dymola.com>
- [24] OpenModelica, Open Source Modelica Consortium (OSMC). Accessed: 2013-12-04. [Online]. Available: <https://www.openmodelica.org/>
- [25] JModelica.org - an extensible Modelica-based open source platform for optimization, simulation and analysis of complex dynamic systems, Moledon AB. [Online]. Available: <http://www.jmodelica.org/>
- [26] PSAT: Power System Analysis Toolbox. [Online]. Available: <http://faraday1.ucd.ie/psat.html>
- [27] K. Clark, N. W. Miller, and J. J. Sanchez-Gasca. Modeling of GE Wind Turbine-Generators for Grid Studies. Version 4.5, Accessed: 2013-12-04. [Online]. Available: <http://goo.gl/x8NBvc>
- [28] FMI Toolbox for Matlab, Modelon AB. Accessed: 2013-12-04. [Online]. Available: <http://www.modelon.com/products/fmi-toolbox-for-matlab/>